



## Penerapan Metode Anaconda Untuk Mendeteksi Orisinalitas Citra Digital

Erwin Saputra Gultom

Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia  
Email [erwiin9@gmail.com](mailto:erwiin9@gmail.com)

### ARTICLE INFO

#### Article History

Received : Nov 24, 2020  
Accepted : Nov 29, 2020  
Published : Nov 30, 2020

### CORRESPONDENCE

Email: [erwiin9@gmail.com](mailto:erwiin9@gmail.com)

### A B S T R A K

Citra memegang peranan sangat penting dalam bentuk informasi visual sebagai salah satu komponen multimedia. Peredaran citra dalam bentuk digital sangat mudah dan cepat baik melalui media sosial atau E-mail (Elektronik-mail). Citra digital lebih dinamis dan dapat dengan mudah di ubah-ubah (editable) baik dari segi bentuk, ukuran, maupun objek yang ada dalam citra digital tersebut. Seiring perkembangan teknologi dan komputer, peningkatan kejahatan komputer pun semakin meningkat terutama dalam manipulasi citra. Banyak cara yang di gunakan orang untuk memanipulasi citra yang dampaknya merugikan orang lain. Orisinalitas citra digital adalah keaslian dari citra tersebut baik dari segi warna, bentuk, objek dan informasi tanpa ada perubahan sedikitpun dari pihak lain. Saat ini banyak citra digital yang beredar di internet yang sudah di manipulasi dan bahkan citra sudah digunakan untuk bahan kecurangan dalam kompetisi, sehingga dibutuhkan suatu metode yang bisa untuk mendeteksi citra tersebut asli atau palsu. Dalam penelitian ini, penulis menggunakan metode Anaconda Hash untuk mendeteksi orisinalitas citra digital, dengan menggunakan metode ini citra yang masih diragukan keasliannya dapat di ketahui citra tersebut asli atau palsu.

**Kata Kunci:** Orisinalitas, Citra, Anaconda

### A B S T R A C T

Image plays a very important role in the form of visual information as one of the multimedia components. The circulation of images in digital form is very easy and fast either through social media or E-mail (Electronic-mail). Digital images are more dynamic and can be easily changed (editable) in terms of shapes, sizes, and objects that are in the digital image. Along with the development of technology and computers, the increase in computer crime has also increased, especially in image manipulation. There are many ways that people use to manipulate images that have a detrimental effect on others. The originality of a digital image is the authenticity of the image in terms of colors, shapes, objects and information without the slightest change from the other party. Nowadays many digital images circulating on the internet have been manipulated and even images have been used for material fraud in the competition, so we need a method that can detect the image is genuine or fake. In this study, the author uses the Anaconda Hash method to detect the originality of digital images, by using this method an image that is still doubtful of its authenticity can be found out that the image is authentic or fake.

**Keywords:** Originality, Image, Anaconda

## 1. PENDAHULUAN

Citra *digital* lebih dinamis dan dapat diubah-ubah (*editable*) baik dari segi bentuk, ukuran, maupun objek yang ada pada citra digital tersebut. Proses pemalsuan terhadap citra *digital* sangatlah mudah dilakukan dengan banyaknya alat bantu berupa perangkat lunak atau aplikasi pengeditan citra yang memungkinkan citra sangat mudah dipalsukan sehingga citra tersebut tidak asli. Hal ini tentunya akan sangat mempengaruhi keanekaragaman citra *digital* yang muncul di masyarakat yang mengakibatkan sulitnya membedakan antara citra yang asli atau palsu. Kemunculan citra-citra digital seperti ini akan menimbulkan permasalahan dalam kehidupan sosial dan merugikan hak milik dari citra tersebut.

Orisinalitas citra digital adalah keaslian dari citra tersebut baik dari segi warna, bentuk, objek dan informasi tanpa ada perubahan sedikitpun dari pihak lain. Saat ini banyak citra digital yang beredar di internet yang sudah di manipulasi dan bahkan citra sudah digunakan untuk bahan kecurangan dalam kompetisi, misalnya lamaran kerja

online, dalam melamar kerja online tentunya yang di butuhkan perusahaan adalah berkas-berkas pelamar dalam bentuk hasil *scan* yang nantinya dikirimkan melalui *e-mail* ke perusahaan tersebut. Dari berkas-berkas lamaran tersebut yang paling di utamakan perusahaan pada umumnya adalah Ijazah, karena disinilah faktor utama seseorang itu lulus atau tidak. Yang pastinya semua orang ingin mendapatkan hasil yang baik dan berharap di terima di perusahaan yang sedang dilamar, maka disinilah seseorang akan sangat mudah melakukan kecurangan, seperti perubahan nilai pada ijazah tersebut.

Banyak tehnik yang digunakan pengguna citra untuk memalsukan citra, seperti tehnik *copy move*, pemotongan, *bluring* dll untuk menghasilkan citra palsu (tidak Orisinal). Dengan cara ini berharap para pengguna citra mendapatkan hasil yang di inginkan dan yang pasti merugikan orang lain. Karena file citra rentan terhadap tindakan pemalsuan, modifikasi dan duplikasi yang tidak berizin yang melanggar hak cipta, sehingga dibutuhkan metode kriptografi untuk mendeteksi orisinalitas citra tersebut. Adapun metode yang digunakan dalam mendeteksi originalitas file citra digital adalah metode *ANACONDA Hash*. Anaconda adalah upaya mendesain fungsi *hash* kriptografi yang mencapai sifat difusi yang baik dari struktur matriks  $4 \times 4$  yang digunakan dalam cipher blok. Anaconda mengacu pada kelas fungsi hash yang mengembalikan output dari berbagai ukuran antara 1 dan 512 bit. Varian yang mengembalikan output ukuran antara 1 dan 256. Fungsi hash Anaconda membutuhkan pesan dengan panjang hingga sekitar  $2^{64}$  bit dan mengembalikan hasil hash  $n$  bit, di mana  $n$  adalah angka antara 1 dan 512. Metode Anaconda mampu mendeteksi orisinalitas citra dengan menemukan perubahan data pada citra digital[1]

## 2. METODOLOGI PENELITIAN

### 2.1 Citra Digital

Citra digital adalah citra yang dapat di olah oleh komputer yaitu gambar pada bidang dua dimensi (baris dan kolom) dimana setiap pasangan indeks baris dan kolom menyatakan suatu titik pada citra. Nilai matriksnya menyatakan nilai kecerahan titik tersebut. Titik-titik tersebut dinamakan sebagai elemen citra, atau *pixel* (*picture element*). Dalam kamus komputer, gambar atau foto di istilahkan sebagai citra yang mempunyai representasi matriks berupa matriks  $C_m \times n = (c_{ij})$ . Citra digital sebagai fungsi intensitas cahaya dua dimensi  $f(x,y)$  dimana  $x$  dan  $y$  menunjukkan koordinat spasial, dan nilai  $f$  pada suatu titik tersebut[2].

### 2.2 Deteksi

Deteksi adalah suatu proses untuk memeriksa atau melakukan identifikasi terhadap sesuatu dengan menggunakan berbagai cara dan teknik tertentu. Deteksi biasanya digunakan untuk menemukan solusi dari masalah, misalnya dalam sistem pendeteksi suatu penyakit, dimana sistem mengidentifikasi masalah-masalah yang berhubungan dengan penyakit yang biasa disebut gejala. Tujuan dari deteksi adalah memecahkan suatu masalah dengan berbagai cara tergantung metode yang diterapkan sehingga menghasilkan sebuah solusi. Selain pada badan radiologi proses pendeteksian selalu digunakan untuk meneliti masalah pada berbagai bidang. Deteksi pada ilmu komputer terkhususnya pada pengolahan citra *digital* selalu digunakan untuk pengambilan garis tepi, median, pixel, noise dll[3].

### 2.3 Orisinalitas Citra Digital

Karya cipta sejatinya sangat berharga bagi para pencipta, namun tidak sedikit juga pihak lain yang melakukan pelanggaran, penjiplakan atau plagiat. Indonesia merupakan salah satu “plagiator”, jika ada suatu karya cipta yang menjadi unggulan atau digemari orang banyak, tentu pihak lain akan berlomba-lomba membuat plagiat atau karya cipta yang menyerupai aslinya[4]. Pencipta atau pemegang hak cipta atas karya steganografi dan program komputer memiliki hak untuk memberikan izin atau melarang orang lain yang tanpa persetujuannya menyewakan ciptaan tersebut untuk kepentingan yang bersifat komersial[4]. Originalitas citra digital adalah keaslian dari file itu sendiri sebelum mengalami perubahan pixel, size dan warna pada pertama kali di publis, serta belum dimanipulasi atau di edit pihak ke dua dst.

### 2.4 Metode Anaconda

Anaconda adalah upaya mendesain fungsi hash kriptografi yang mencapai sifat difusi yang baik dari Struktur matriks  $4 \times 4$  yang digunakan dalam cipher blok Square dan Rijndael[5]. Nama Anaconda mengacu pada kelas fungsi hash yang mengembalikan output dari berbagai ukuran antara 1 dan 512 bit. Varian yang mengembalikan output ukuran antara 1 dan 256 bit, semuanya menggunakan metode yang sama, hanya nilai awal dan jumlah pemotongan yang terjadi pada akhirnya berbeda. Hasil akhirnya adalah fungsi hash yang memiliki kemiripan dengan Rijndael dan Serpent secara bersamaan. Sederhana dan elegan[6].

### 2.5 Spesifikasi Anaconda

Fungsi hash Anaconda  $H$  membutuhkan pesan dengan panjang hingga sekitar  $2^{64}$  bit dan mengembalikan hasil hash  $n$  bit, di mana  $n$  adalah angka antara 1 dan 512. Anaconda mengasumsikan pemesanan *byte big-endian*. Ini menerapkan fungsi kompresi  $f : \{0, 1\}^{512} \times \{0, 1\}^{512} \rightarrow \{0, 1\}^{512}$  dalam mode standar *Merkle-Damgård* [7]. Hasil dari aplikasi akhir dari fungsi kompresi dipotong ke nilai  $n$ - bit. Fungsi kompresi  $f$  beroperasi dengan status 16 kata, yang terlihat sebagai  $4 \times 4$  matriks kata-kata (seperti dalam Rijndael). Keadaan  $A = a_0 || \dots || a_{15}$  dilihat sebagai matriks berikut.

$$A = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} \\ a_{12} & a_{13} & a_{14} & a_{15} \end{bmatrix}.$$

Setiap kata berukuran 64 bit. Misalkan  $W = \{0,1\}^{64}$ . Fungsi kompresi mengambil blok pesan 512-bit dan nilai rantai 512 bit, membentuk keadaan 1024-bit yang dilihat sebagai matriks di atas, dan menerapkan sejumlah putaran ke status. Keadaan akhir kemudian dipotong menjadi 512 bit. Fungsi putaran sekarang dijelaskan.

## 2.6 Fungsi Bilangan bulat

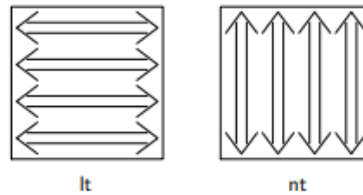
Fungsi bilangan bulat  $p: W^{16} \rightarrow W^{16}$  adalah permutasi. Ini menerapkan dua transformasi berbeda. Dengan menggunakan transformasi linear  $lt: W^4 \rightarrow W^4$ , dan transformasi non-linear  $nt: W^4 \rightarrow W^4$  dan akan kembali ke bagaimana transformasi ini didefinisikan dalam karakter a. Fungsi putaran beroperasi pada keadaan  $A = a_0 \parallel \dots \parallel a_{15}$  sebagai berikut.

$a_{15}$	$\leftarrow a_{15} \oplus 1$
$(a_0, a_1, a_2, a_3)$	$\leftarrow lt(a_0, a_1, a_2, a_3)$
$(a_7, a_4, a_5, a_6)$	$\leftarrow lt(a_7, a_4, a_5, a_6)$
$(a_{10}, a_{11}, a_8, a_9)$	$\leftarrow lt(a_{10}, a_{11}, a_8, a_9)$
$(a_{13}, a_{14}, a_{15}, a_{12})$	$\leftarrow lt(a_{13}, a_{14}, a_{15}, a_{12})$
$(a_0, a_4, a_8, a_{12})$	$\leftarrow nt(a_0, a_4, a_8, a_{12})$
$(a_1, a_5, a_9, a_{13})$	$\leftarrow nt(a_1, a_5, a_9, a_{13})$
$(a_2, a_6, a_{10}, a_{14})$	$\leftarrow nt(a_2, a_6, a_{10}, a_{14})$
$(a_3, a_7, a_{11}, a_{15})$	$\leftarrow nt(a_3, a_7, a_{11}, a_{15})$

**Gambar 1.** Fungsi putaran bulat

Sumber : Thomsen, Soren Steffen 2008 : 2[6]

Pada gambar diatas pertama lapisan linear diterapkan pada setiap baris, dan kemudian non-lapisan linier diterapkan ke setiap kolom. Lapisan linier memberikan difusi baik pada bit-level dan pada level kata. Lapisan non-linear adalah S-box 4-bit dalam *bitslice* mode, yang menyediakan difusi pada level kata (berdasarkan kolom). Perhatikan bahwa urutan kata-kata input ke fungsi  $lt$  digeser untuk setiap baris, dan itu paling tidak signifikan dari 15 dibalik di awal. Langkah-langkah ini untuk memperkenalkan asimetri.



**Gambar 2.** Efek transformasi  $lt$  dan  $nt$ .

Sumber : Thomsen, Soren Steffen 2008 : 3[6]

## 2.7 Transformasi Linier It

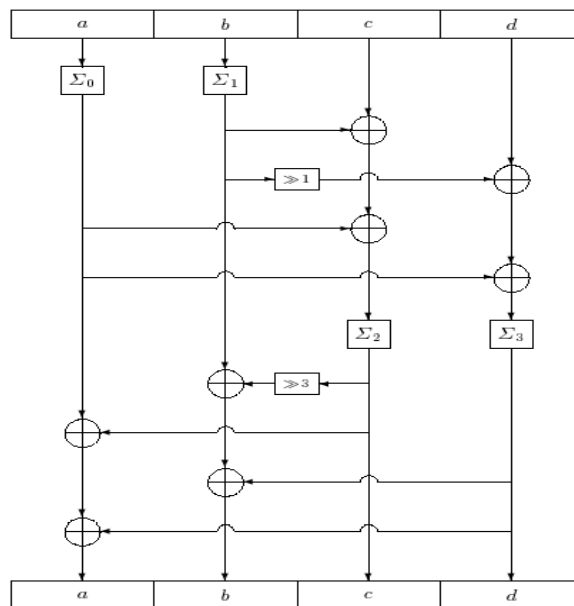
Dalam definisi  $lt$ , empat transformasi yang beroperasi pada satu kata masing-masing diterapkan. Biarkan ini menjadi  $\Sigma_i$ ,  $0 \leq i < 4$ . Didefinisikan sebagai berikut (di mana  $a \ll r$  berarti kiri yang diputar oleh posisi  $r$ ).

$$\begin{aligned} \Sigma_0(a) &\leftarrow a \ll 1 \oplus a \ll 20 \oplus a \ll 24 \\ \Sigma_1(a) &\leftarrow a \ll 13 \oplus a \ll 22 \oplus a \ll 60 \\ \Sigma_2(a) &\leftarrow a \ll 30 \oplus a \ll 50 \oplus a \ll 63 \\ \Sigma_3(a) &\leftarrow a \ll 24 \oplus a \ll 51 \oplus a \ll 54 \end{aligned}$$

Diberikan empat kata input ( $a, b, c, d$ ),  $lt$  didefinisikan sebagai berikut.

$$\begin{aligned} a &\leftarrow \Sigma_0(a) \\ b &\leftarrow \Sigma_1(b) \\ c &\leftarrow a \oplus b \oplus c \\ d &\leftarrow a \oplus (b \gg 1) \oplus d \\ c &\leftarrow \Sigma_2(c) \\ d &\leftarrow \Sigma_3(d) \\ a &\leftarrow a \oplus c \oplus d \\ b &\leftarrow b \oplus (c \gg 3) \oplus d \end{aligned}$$

$a \gg r$  berarti bergeser ke kanan dengan posisi  $r$  bit (karenanya, bit  $r$  yang paling kiri dari  $a \gg r$  adalah nol) pada gambar berikut.



**Gambar 2.** Transformasi linier It  
Sumber : Thomsen, Soren Steffen 2008 : 4[6]

## 2.8 Transformasi non-linier nt

Seperti yang disebutkan, transformasi non-linear nt adalah S-box 4-bit dalam mode *bitslice*. S-box  $S$  yang digunakan untuk nt didefinisikan sebagai berikut.

$x$ :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S(x)$ :	2	9	8	3	11	5	7	14	12	15	1	4	6	0	10	13

**Gambar 3.** S-box 4-bit dalam mode *bitslice*

S-box ini, secara konseptual, diterapkan sebagai berikut. Tempatkan bit  $j$  dari kata input ke-  $i$  (dihitung dari 0) pada posisi  $(i, j)$  pada  $4 \times 64$  matriks bit. Terapkan kotak-S untuk masing-masing 64 kata 4-bit yang di definisikan oleh kolom matriks ini, di mana bit di baris atas berada bit paling signifikan. Kemudian, petakan matriks kembali ke empat kata 64-bit. Dalam prakteknya, the Aplikasi S-box dapat dilakukan melalui operasi logis pada empat kata 64-bit[1].

## 3. HASIL DAN PEMBAHASAN

Pendeteksian Orisinalitas citra digital sangat diperlukan karena seringkali citra mengalami manipulasi, misalnya perubahan warna dan dan objek pada citra yang sudah di edit melalui aplikasi pengolahan gambar seperti *photoshop*, *coreldraw* dll. Ini menjadi permasalahan ketika citra dipublikasikan ke media sosial dan elektronik tanpa seijin dari pihak pertama, dapat merugikan hak cipta seseorang karena citra sudah mengalami perubahan. Proses deteksi orisinalitas citra bertujuan untuk mempermudah langkah analisis citra palsu atau asli dan informasi citra secara detail. Untuk mengatasi citra yang bermasalah tersebut maka dibutuhkan teknik untuk mendeteksinya. Adapun metode yang digunakan untuk mendeteksi citra *digital* ialah metode *Anaconda Hash*. Deteksi pemalsuan citra dilakukan terhadap citra ijazah. Langkah-langkah untuk mendeteksinya adalah memasukkan sidik pesan fungsi *Hash* untuk mewujudkan layanan keutuhan data. Misalnya  $M$  merupakan pesan dan  $h$  adalah fungsi *Hash*, maka  $y=h(M)$  disebut dengan sidik pesan  $x$  atau sering juga disebut *message digest*. Sebuah *message digest* umumnya berukuran pendek, yaitu sekitar 160 bit. Dengan menggunakan sebuah fungsi *hash*  $h$ . Sebelum pesan  $M$  disebarkan/dikirimkan sebuah *message digest*  $y_{lama} = h(M)$  disimpan sebagai acuan. Misalnya didapatkan kembali  $M'$  setelah disebarkan/dikirim apabila ingin menguji apakah  $M=M'$  hitung kembali *message digest* baru  $y_{baru} = h(M')$  disimpulkan pesan tidak berubah bila  $y_{lama} = y_{baru}$ . Setelah dilakukan keutuhan data pada citra maka langkah selanjutnya adalah menghitung atau mengukur perbedaan antara citra sebelum dan sesudah di deteksi dengan memanfaatkan metode anaconda yaitu teknik transformasi linear It dan transformasi non linear nt, maka dapat diukur perbedaan antara citra sebelum dan sesudah dideteksi.

### 3.1 Penerapan Metode *Anaconda*

Metode *Anaconda* merupakan suatu model yang dapat mengembalikan output dari berbagai ukuran antara 1 dan 215 bit. Pada awal proses input yang telah dikopikan ke dalam *state* akan mengalami transformasi linear It dan transformasi non-linear nt. Anaconda mampu mendesain fungsi hash kriptografi dengan menentukan nilai biner dari setiap matriks yang telah diambil. Berikut telah disiapkan citra ijazah yang mengalami perubahan seperti penggantian nama pada ijazah asli dengan dimensi 2480 x 3508 *pixel*, dengan format JPG. Berikut tampilan yang dijadikan sampel pengujian. Adapun *script* matlab untuk memasukkan dan menampilkan gambar adalah sebagai berikut :

```
img = imread('c:\foto\ijazah.jpg') ;  
Figure, imshow(img) ;
```

Nilai piksel 4 x 4 RGB (Red, Green, Blue) diambil dari sudut bagian kiri atas seperti yang digambarkan dibawah ini, adapun *script* matlab untuk mengambil nilai matriks gambar adalah sebagai berikut :



#### Gambar 4. Citra Ijazah

Kemudian untuk mendapatkan nilai tengah, nilai RGB dari setiap masing-masing matriks tersebut di jumlahkan secara keseluruhan dan hasilnya di bagi tiga sehingga hasilnya sebagai berikut :

$$A = \begin{bmatrix} 245 & 245 & 245 & 245 \\ 244 & 244 & 245 & 245 \\ 243 & 243 & 244 & 244 \\ 243 & 243 & 243 & 243 \end{bmatrix}$$

**Tabel 1.** Nilai biner dari matriks

11110101	11110101	11110101	11110101
11110100	11110100	11110101	11110101
11110011	11110011	11110100	11110100
11110011	11110011	11110011	11110011

**Tabel 2.** *Padding* pada Nilai biner

[illegible]

Kemudian memasukkan persamaan  $A = [a_0 | \dots | a_{15}]$  untuk memberikan variabel pada setiap baris dan kolom pada matriks.

$$\begin{aligned} a_0 &= 11110101 \ 11110101 \ 11110101 \ 11110101 \\ a_1 &= 11110100 \ 11110100 \ 11110101 \ 11110101 \\ a_2 &= 11110011 \ 11110011 \ 11110100 \ 11110100 \\ a_3 &= 11110011 \ 11110011 \ 11110011 \ 11110011 \\ a_4 &= 10000000 \ 00000000 \ 00000000 \ 00000000 \\ a_5 &= 00000000 \ 00000000 \ 00000000 \ 00000000 \\ a_6 &= 00000000 \ 00000000 \ 00000000 \ 00000000 \end{aligned}$$

$a_7 = 00000000\ 00000000\ 00000000\ 00000000$   
 $a_8 = 00000000\ 00000000\ 00000000\ 00000000$   
 $a_9 = 00000000\ 00000000\ 00000000\ 00000000$   
 $a_{10} = 00000000\ 00000000\ 00000000\ 00000000$   
 $a_{11} = 00000000\ 00000000\ 00000000\ 00000000$   
 $a_{12} = 00000000\ 00000000\ 00000000\ 00000000$   
 $a_{13} = 00000000\ 00000000\ 00000000\ 00000000$   
 $a_{14} = 00000000\ 00000000\ 00000000\ 00000000$   
 $a_{15} = 00000000\ 00000000\ 00000000\ 10000000$

# 1. Transformasi linier It

$$\begin{aligned}
 a_{15} &= a_{15} \oplus 1 \\
 &= 00000000\ 00000000\ 00000000\ 10000000 \\
 &\quad \underline{00000000\ 00000000\ 00000000\ 00000001} \oplus \\
 &= 00000000\ 00000000\ 00000000\ 10000001
 \end{aligned}$$

$$(a_0, a_1, a_2, a_3) = \text{It}(a_0, a_1, a_2, a_3)$$

$$\Sigma_0(a) \leftarrow a \lll 1 \oplus a \lll 20 \oplus a \lll 24$$

$$\begin{aligned}
 a \lll 1 &= 11101011\ 11101011\ 11101011\ 11101011 \\
 &\quad 11101001\ 11101001\ 11101011\ 11101011 \\
 &\quad 11100111\ 11100111\ 11101001\ 11101001 \\
 &\quad 11100111\ 11100111\ 11100111\ 11100111
 \end{aligned}$$

$$\begin{aligned}
 a \lll 20 &= 01011111\ 01011111\ 01001111\ 01001111 \\
 &\quad 01011111\ 01011111\ 00111111\ 00111111 \\
 &\quad 01001111\ 01001111\ 00111111\ 00111111 \\
 &\quad 00111111\ 00111111\ 01011111\ 01011111
 \end{aligned}$$

$$\begin{aligned}
 a \lll 24 &= 11110101\ 11110100\ 11110100\ 11110101 \\
 &\quad 11110101\ 11110011\ 11110011\ 11110100 \\
 &\quad 11110100\ 11110011\ 11110011\ 11110011 \\
 &\quad 11110011\ 11110101\ 11110101\ 11110101
 \end{aligned}$$

$$\begin{aligned}
 \text{Xor}(\oplus) &= 11101011\ 11101011\ 11101011\ 11101011 \\
 &\quad 01011111\ 01011111\ 01001111\ 01001111 \\
 &\quad \underline{11110101\ 11110100\ 11110100\ 11110101} \oplus \\
 \Sigma_0(a) &= 01000001\ 01000000\ 01010000\ 01010001
 \end{aligned}$$

$$\begin{aligned}
 &11101001\ 11101001\ 11101011\ 11101011 \\
 &01011111\ 01011111\ 00111111\ 00111111 \\
 &\underline{11110101\ 11110011\ 11110011\ 11110100} \oplus \\
 \Sigma_0(a) &= 01000011\ 01000101\ 00100111\ 00100000
 \end{aligned}$$

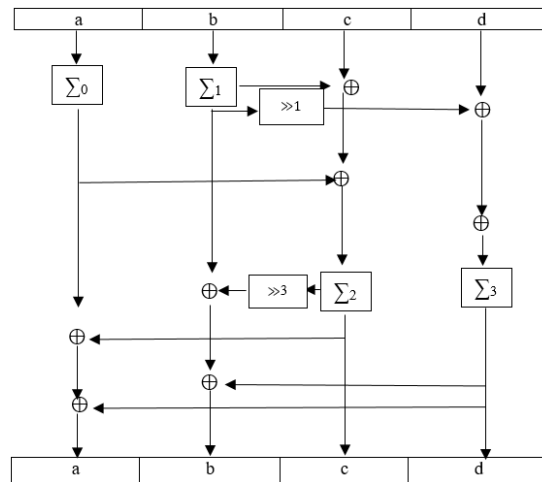
$$\begin{aligned}
 &11100111\ 11100111\ 11101001\ 11101001 \\
 &01001111\ 01001111\ 00111111\ 00111111 \\
 &\underline{11110100\ 11110011\ 11110011\ 11110011} \oplus \\
 \Sigma_0(a) &= 01011100\ 01011011\ 00100101\ 00100101
 \end{aligned}$$

$$\begin{aligned}
 &11100111\ 11100111\ 11100111\ 11100111 \\
 &00111111\ 00111111\ 01011111\ 01011111 \\
 &\underline{11110011\ 11110101\ 11110101\ 11110101} \oplus \\
 \Sigma_0(a) &= 00101011\ 00101101\ 01001101\ 01001101
 \end{aligned}$$

$$\begin{aligned}
 \Sigma_0(a) &= 01000001\ 01000000\ 01010000\ 01010001 \\
 &01000011\ 01000101\ 00100111\ 00100000 \\
 &01011100\ 01011011\ 00100101\ 00100101 \\
 &00101011\ 00101101\ 01001101\ 01001101
 \end{aligned}$$

$$\begin{aligned}
 (a_7, a_4, a_5, a_6) &= \text{It}(a_7, a_4, a_5, a_6) \\
 \Sigma_1(a) &\leftarrow a \lll 13 \oplus a \lll 22 \oplus a \lll 60 \\
 (a_{10}, a_{11}, a_8, a_9) &= \text{It}(a_{10}, a_{11}, a_8, a_9) \\
 \Sigma_2(a) &\leftarrow a \lll 30 \oplus a \lll 50 \oplus a \lll 63 \\
 (a_{13}, a_{14}, a_{15}, a_{12}) &= \text{It}(a_{13}, a_{14}, a_{15}, a_{12}) \\
 \Sigma_3(a) &\leftarrow a \lll 24 \oplus a \lll 51 \oplus a \lll 54
 \end{aligned}$$

Dari hasil a,b,c,d di atas kemudian melakukan transformasi linier pada setiap baris sebagai nilai input.



**Gambar 5.** Transformasi linier

Setelah semua nilai  $a, b, c, d$  melakukan transformasi linier It maka diperoleh *Output* dari masing –masing variabel. Kemudian hasil output pada transformasi linier, dilakukan perputaran ke kiri dengan cara fungsi permutasi:

1.  $a \leftarrow \Sigma_0(a)$   
 $b \leftarrow \Sigma_1(b)$   
 $c \leftarrow a \oplus b \oplus c$   
 $d \leftarrow a \oplus (b \ll 1) \oplus d$
2.  $c \leftarrow \Sigma_2(c)$   
 $d \leftarrow \Sigma_3(d)$   
 $a \leftarrow a \oplus c \oplus d$   
 $b \leftarrow b \oplus (c \gg 3) \oplus d$

Setelah dilakukan semua rumus pergeseran di atas maka diperoleh nilai  $a, b, c, d$  sebagai berikut.

$a =$  10010010 10100000 10001100 10110110  
 10010010 00110011 10100011 01101100  
 10100100 00101011 11100111 10100111  
 10101110 11110100 01100000 01100001

$b =$  11010011 11000000 11001100 11100111  
 11111100 00010011 00110100 00110100  
 01110010 10001010 11001010 10110010  
 10111101 00110000 00110001 10110000

$c =$  01000001 01100000 01000000 01010001  
 01101011 00101101 10000011 11000100  
 01010111 11010000 01000001 10000001  
 10000111 01001000 11100100 11100100

$d =$  10010010 10101000 10001000 10110110  
 10011110 01011011 00000111 10001000  
 10101111 10100000 10000011 00000011  
 00001110 10010001 11001001 11001000

Dari hasil pergeseran ke kanan maka setiap baris dari hasil  $a, b, c$  dan  $d$  tersebut di ubah ke hexadesimal untuk memperoleh nilai hash.

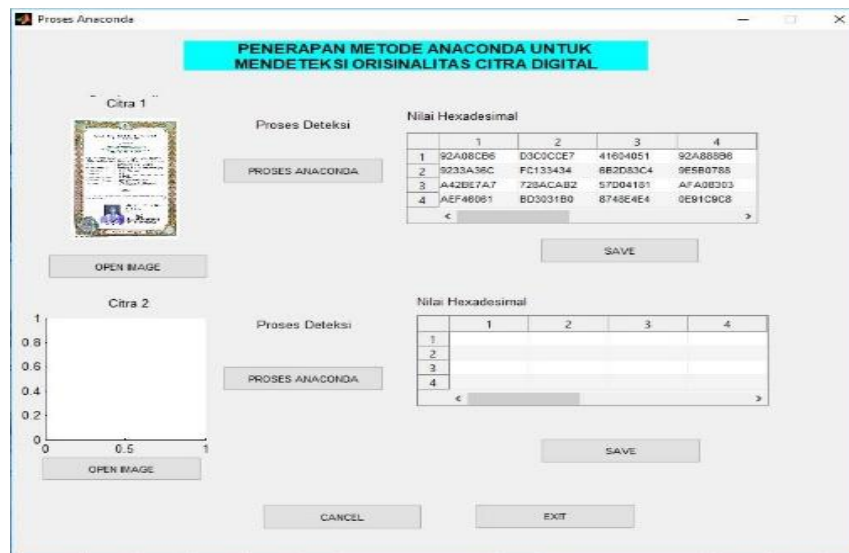
**Tabel 3.** Nilai Hash hexadesimal

$a$	$b$	$c$	$d$
92A08CB6	D3C0CCE7	41604051	92A888B6
9233A36C	FC133434	6B2D83C4	9E5B0788
A42BE7A7	728ACAB2	57D04181	AFA08303
AEF46061	BD3031B0	8748E4E4	0E91C9C8

### 3.2 Pengujian

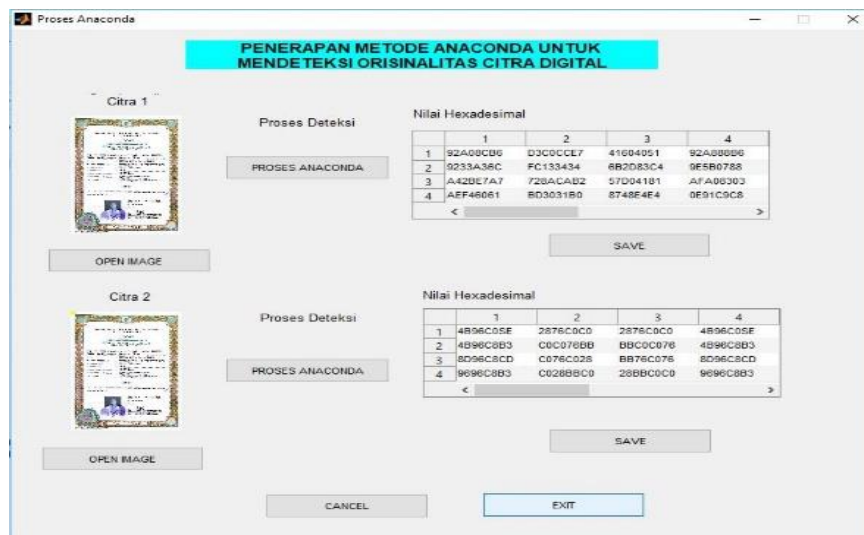
Implementasi deteksi citra asli atau palsu menggunakan metode *Anaconda* digunakan pada satu *set* unit komputer multimedia. Sesuai dengan perancangan input, proses dan ouput aplikasi ini dapat dilakukan sesuai dengan kebutuhan perangkat keras dan perangkat lunak. Hasil merupakan suatu proses dari penyelesaian suatu masalah yang mana masalah ini adalah untuk deteksi orisinalitas citra digital. Tampilan pengujian merupakan proses yang bertujuan untuk memastikan apakah semua fungsi sistem bekerja dengan baik sesuai dengan perancangan dan mencari kesalahan yang mungkin terjadi pada sistem yang dibuat. Pengujian deteksi orisinalitas citra digital ini dilakukan pada citra ijazah hasil scan.

Ini merupakan tampilan dari hasil deteksi, ketika gambar sudah di *input* kemudian dilakukan proses *Anaconda*, maka hasil akan terlihat disamping gambar tersebut berupa nilai hexadesimal di dalam tabel.



Gambar 6. Hasil deteksi nilai hexa Ijazah 1

kemudian lanjut untuk menginput gambar kedua. Dengan langkah yang sama yaitu melalui tombol open image, kemudian dilakukan proses Anaconda untuk menemukan nilai hexadesimal yang baru, sehingga tampil pada gambar berikut.



Gambar 7. Hasil deteksi nilai hexa Ijazah 2

Setelah dilakukan pendeteksian Ijazah pertama dan kedua, dan nilai hexadesimal pada gambar telah ditemukan, maka user tinggal membedakan nilai hexadesimal dari kedua gambar tersebut.

## 4. KESIMPULAN

Adapun kesimpulan yang penulis dapatkan dari hasil penelitian, yaitu mendeteksi orisinalitas citra menggunakan metode *Anaconda* dilakukan dengan mengambil nilai input 4x4 dari citra itu sendiri dan penambahan *bit* untuk melakukan transformasi linier *It* dan transformasi non-linier *nt*. Penerapan metode *Anaconda* untuk mendeteksi orisinalitas citra *digital* mampu mendapatkan nilai hexadesimal yang berbeda antara citra asli dengan citra palsu. Dari hasil pengujian metode *Anaconda* didapatkan hasil citra yang asli atau palsu setelah di proses dengan membandingkan nilai hexadesimal.

## REFERENCES

- [1] S. Steffen, "The Anaconda hash functions The A NACONDA hash functions Søren Steffen Thomsen," p. 11, 2008.
- [2] D. Putra, *Pengolahan Citra Digital*, 1st ed. Yogyakarta: ANDI, 2010.
- [3] M. S. B. Alvin A. Arens, Randal J. Elder, "Pengertian Deteksi & Disleksia (2012:4)," vol. Ii, B A B, pp. 11–43, 2012.
- [4] S. Budi, "ORISINALITAS SUATU KARYA CIPTA (1)." Pustaka Magister, Semarang, p. 91, 2011.
- [5] V. R. 2. J. Daemen, L. R. Knudsen, "The Block Cipher Square.," pp. 149–165, 1997.
- [6] S. S. Thomsen, M. No, and S. S. Thomsen, "The A NACONDA hash functions The Anaconda hash functions," 2008.
- [7] 4.I. Damgård, . "A Design Principle for Hash Functions," vol. 435, pp. 416–427, 1990.